下载APP

27 | 答疑篇：什么样的技术观能够更快成长？

2020-07-17 臧萌

职场求生攻略 进入课程 >

讲述：臧萌
时长 12:11 大小 11.17M

你好，我是臧萌。转眼之间，咱第四个模块也结束了。在这个模块里，我们围绕着技术，聊了聊程序员的技术领地、我们程序员应该如何看待技术、如何从主要做开发的程序员成长为软件架构师、以及如何不掉进软件系统集成的那些坑。

在一开始，我们只要关心技术就可以了，打好技术基础，养成学习技术的习惯。紧接着，我们要开始审视自己看待技术的观点，也就是技术观。因为随着我们程序员的成长，工作的重心将开始从单纯的技术，转向如何用技术服务业务。

在这个过程中，如果你打算继续走技术路线，很大的一种可能就是工作重心继续上移，自己思考技术服务业务的过程中，逐渐培养自己做软件架构的能力，进而成为一名软件系统架构师。

软件架构师需要哪些能力？

软件系统架构师，要有自己的经验积累和独到的眼光，要能够很好地认识问题、对问题建模、解决问题。除此之外，还要考虑的一点就是软件系统的集成。因为软件系统的集成，才是对一个软件系统的终极大考，也是对架构师综合能力的大考。一个软件架构看上去很美的系统，如果不能很好地落地集成，最后的结果只会是叫好不叫座，被束之高阁。

你如果能够走通这些关键节点，在我看来就已经是一个软件研发的全能人才了。软件研发全能人才更多地是从个人综合能力上来讲，而我们平时说的全栈工程师，更多地是从技术层面来谈。技术是武器，使用哪种武器解决问题，肯定会影响最终的架构。不过在架构之上，理解问题、分析问题、解决问题的能力，这个和具体的技术是没有太大关系的。

所以，我更愿意建议程序员，不要把自己的视野局限在技术层面，而是要把视野扩展到从看到问题，到最终解决问题这么一个全过程。这也是整个技术成长篇里，我想表达的一个非常重要的观点。

必须补充一点的是，并不是说做了软件系统架构师，就不需要持续学习技术了。软件系统架构师也必须持续不断地更新自己的技术库，维持甚至扩张自己的技术领地。这样才能够让自己做出适合当下甚至未来发展的架构，可以进一步可以减少系统集成的风险。技术是武器，熟悉武器的发展，指挥起打仗来心里才能有底气，才能灵活应对。

这部分里，很多同学提出了非常多有价值的问题，也有同学分享了自己的心得体会，这里我们一起来看一下。

修炼内功，厚积薄发

关于技术舒适区，**@牛牛** 同学分享了自己的感受。开始的时候，学习很痛苦，等熬过这段时间，就会有各种小惊喜，持续不断地形成正反馈、就越来越有学习的动力了。小惊喜来自于随着知识的融会贯通，之前不懂的事情，一下就懂了。

的确是这样的，尤其是数据结构、 算法、操作系统、网络、计算机系统结构这些内功，虽然没有立竿见影的效果，但是会对你还没学的东西都有加成。你会发现内功一旦深厚了，学习其他东西会很快。

举个简单的例子，HBase 里用一个叫做 Bloom Filter 的数据结构去做过滤，Bloom Filter 的特点是如果这个值存在，Bloom Filter 绝对会返回 true，但是如果某个值不存在，Bloom Filter 可能会返回 true，也可以返回 false。它里面用了一种非常精妙的设计，可以在可控的内存占用下，达到预期的判定准确率。如果不事先学习，肯定会让自己学起来阻力重重。但是如果之前就学会了，那看到的时候就只需会心一笑。

学好基础，以后学习就一帆风顺。基础不行，学习就是步步维艰。

搭建骨架，稳步前进

@pyhhou 同学对如何构建自己的技术骨架比较感兴趣。我觉得这是一个很好的问题。确实技术骨架不是一开始就能搞起来的，尤其是在自己的知识存量还没有形成规模效应，可以让我们触类旁通的时候。

我从开荒的状态说起吧，以 Spring 为例子。我们首先肯定是要学好 Java，明白对象、类、方法、多态、继承等核心概念。对于 Spring 来说，慢慢还要知道注解、反射、代理这些 Spring 会重点用到的知识。

然后就是学 Spring，和学一般的技术一样，你需要找个趁手的书或者资料，一以贯之地啃完。可以是我们的课程，也可以是官方文档。这个过程，开始更多的是比着葫芦画瓢，先看效果。

接着，你就可以发挥自己的想象力来探索了。这时候的最好的方式就是多写程序，想到什么就拿出来写写、练练，验证自己的想法。这一步，就是在慢慢构建你自己的技术骨架了。

所谓骨架，就是藏在表面之下的东西，虽然看不到，但是如果没有它们支撑的话，表面的东西就不牢固。就拿 Spring 来说，你要能知道这些 bean 是如何被描述、被创建、被组装的。做到这些之后，当出了某种异常，你就很容易能找出是哪个步骤的问题，马上去修正。

如果能做到这些，让你自己写一个 IoC 的框架，也不是难事。当然，并不是说一定要做的像 Spring 这么全面，可能只是一个微小的内核，但是相信我，如果你自己做一遍、思考一遍，你会对 Spring 的认识更近一层。当你把自己的设计和实现与 Spring 做对比时，会记得更牢固，印象更深刻。这个步骤，可以帮助你完整地构建关于这个技术的骨架。

当然，也不是说每个东西都要自己照着写一遍。即使不去自己做一遍，也要有意识地在脑子里思考一下，如果是你自己做，你会如何设计、如何架构、遇到想不通的地方去仔细看看现有的设计，能够有很多收获。

@Sdylan 同学也分享了自己构架技术骨架的方式。虽然方式不同，但和我的划分也有异曲同工之妙。

其实每种技术都有侧重性，有些技术可能也需要这几种模型之外的模型。比如偏向数据存储和处理的技术，重点考虑它的架构，然后是数据模型和线程模型。把这几个搞清楚了，就知道这个玩意是怎么转的，出了问题可能是哪里的问题，也能知道它是否适合一个场景。

其实每个人都有自己的习惯，重要的是要有个技术骨架，这样才能把知识整合在一起，才能系统地学会新领域的知识。否则就容易什么东西都只会个表面，容易浮躁。

服务业务，不忘初心

在技术观部分，Sdylan 同学提到，空谈技术，不顾业务，只会误了技术误了业务。技术本来就是为了解决问题而生的，若脱离了问题。就会变得毫无价值，生命周期也是短的。只有落地，随着业务发展才能长久。对个人而言，用到必须里外吃透，做的需求至少要理解到为啥做需求，解决什么业务痛点。

关于生命周期，可以分享一个我的看法。其实做技术久了会发现，技术过几年大概率会扔掉，而业务反而不容易扔掉，积累起来越来越值钱。

其实我们程序员容易忽视业务，或者说容易钻到技术里出不来，也是有客观原因的。就是做技术需要全心全意的投入，就是需要你钻进去。这时候，就容易忽略我们的初心：服务业务。所以我们程序员可以时不时地，从技术的世界里钻出来，重新想想业务是什么，我们钻的深度和方向，是否还是为了业务服务。

从业务来，到业务去

在系统架构部分，有同学表示好的架构都是从业务问题来的，然后再抽象出来。一个很好的例子就是 Kafka。看完此文章后，也开始慢慢理解公司设计出来的内部架构，以前觉得这玩意不就是 Spring 系列，值得这么整么。其实发现是自己对业务的理解过于狭窄，随着

接触的业务多了，开始慢慢明白为啥这么做。总之，架构师首先是业务架构师，本质上是业务与架构的 tradeoff，关键是对问题思考和抽象程度。

这个总结很到位。所以你也理解为什么软件工程师有饭吃吧。同样是造轮子，在我看来，软件工程师不是为了重复造轮子，因为对于每辆飞速行驶的车来说，轮子上各种细节的雕琢，都可能会对业务带来巨大的提升。

这辆业务的车是在沙漠里开，还是在湿路上开，还是在山路上开，都会对轮子提出非常具体的要求。甚至沙子是细沙还是粗沙，都不一样。所以只要全世界的业务还都没挪到软件系统上，只要业务还在发展，软件工程师就一直有饭吃。

当然，重新造轮子，得到最大锻炼的就是架构师。架构师要珍惜珍惜再珍惜每一个重新造轮子的过程，深入理解公司业务的需求，构建业务模型，打造能帮助公司飞驰的轮子。解决实际的问题，是锻炼的绝佳机会。这种公司花钱出人让人得到实践锻炼的机会，遇到了做梦都能笑醒。

保持数据警醒度

关于系统集成，很多同学都提到了 / 打印外部系统传来的参数。这个确实是一个很重要的点。确实系统集成的很大一部分就是数据。比如接入别人的数据，读取别人写的数据库，把数据传给别人等等。

为什么数据这么容易出问题呢？因为数据交互就涉及到数据的序列化和反序列化，数据的封装和传输。其中任何一个环节都可能造成数据的错误或丢失。

比如说最常见的 http 协议。http header 虽然可以放一些数据，但是它是有长度限制的。而且不同的服务器对这个长度限制的默认设置也不一样。所以在系统集成时，任何外部的数据都不要信，同时，发给外部的数据，自己也记录一下。

总结

在这一篇里，我们聊了技术和架构，核心是程序员的成长。虽然没有涉及到具体的技术知识，但跳出纯技术的角度来看程序员这份工作，知道我们为何学技术、知道如何用技术、思考如何用技术给自己的职业生涯助力，也是非常重要的事情。

这一篇的内容，是按照由浅入深来安排的。侧重点分别是为何要持续学习技术，如何看待技术和工作，如何培养自己架构师的能力，以及如何培养自己将系统落地的能力。你可以参考自己当前的状态，思考之前是否有不足，也希望能对你日后的职业发展有所帮助。

以上就是答疑篇的内容，不知道有没有解决你的困惑呢？

欢迎你在评论区与我交流一下你的想法，也欢迎把这篇文章分享给你的朋友或者你的同事，一起来交流一下。

提建议

更多课程推荐

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 26 | 系统集成：为什么最容易出问题的是系统集成？

下一篇

精选留言 (2)

写留言

Sdylan
2020-07-17

意犹未尽，专栏就要结束。舍不得呀。

展开

作者回复: 哈哈，完结了我们还可以在评论区继续聊

3

Eden Ma
2020-07-18

下周就完结了哈哈 有不少收获

展开

作者回复: 很高兴对你有帮助😄